RemoDAQ-8188END 模块

用户手册

北京鼎升力创技术有限公司

目　录

1 概述

在写字楼、工厂、住宅小区里面 Ethernet 是基础设施之一。你可以远程监控或者很便利的访问服务器，工业上使用 Ethernet 最吸引人的就是可以方便的收集到所需资料信息。

RemoDAQ-8188END 系列实现Ethernet/Internet 应用的嵌入式控制器，它可以替代PC或PLC 应用于恶劣环境境，结合RemoDAQ-8188END系列模块完成远程数据监控任务，RemoDAQ-8188END系列提供一个10BASE-T 网口及NE2000 网路控制器，连接器为RJ-45。它配有一个512KB的静态RAM和512KB Flash Memory的80188-40处理器，提供RS232和RS485端口。RemoDAQ-8188END就像RS232与Ethernet的转接器一样。中央计算机在RemoDAQ-8188END的帮助下通过Ethernet连接所有的RS232端口，RS232设备将连接最近的网络HUB和中央计算机，连接所有的设备以后，软件应用将会访问他们。通常，一个TCP/IP装置应用程序比COM端口应用程序要难以显示，“VSP”或者是“VIRCOM”可以从TOPSCCC公司或者EASYUNET公司的网站或光盘上得到，它能用来安排网络协议阶层，安排主机有效的、实际安装在RemoDAQ-8188END上的COM端口。因此，软件应用方面COM端口的装置和旧的COM端口装置仍被使用。它的优势就决定了它可以为软件的发展提供一个简单的界面，也可以保持现有的旧系统没有程序上的修改。

随RemoDAQ-8188END提供函数库支持TCP/IP 协议及Web Server，TCP/IP 库包括：TCP, UDP, IP, ICMP, ARP, RARP。

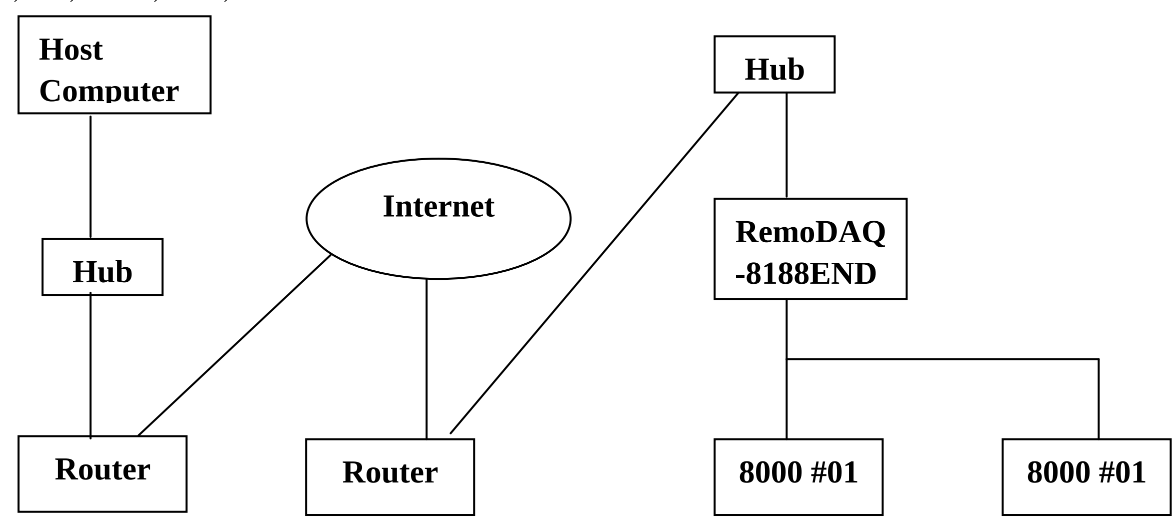


图 1.1 RemoDAQ-8188ND 网络简易图

COM 端口的默认值				
	波特率	数据位	奇偶	停止位
COM1	57600	8	无	1
COM2	9600	8	无	1
COM3	9600	8	无	1
COM4	9600	8	无	1
COM5	9600	8	无	1
COM6	9600	8	无	1
COM7	9600	8	无	1
COM8	9600	8	无	1

2 系统

RemoDAQ-8188END 是网络端的工业控制器，含有 80188CPU，支持下列函数，包括所有的硬件函数库，关于库的应用，详见第五章。

1. 512K：SRAM 和 FALSH
2. 内有实时时钟(RTC)，NVRAM，EEPROM
3. 内有 5 位数字显示器
4. 1 个 RS485 端口
5. 1 个 5-wire 控制 RS232 端口
6. 6 个 RS232 端口
7. 1 个系统定时器（中断号 no.08）
8. 1 个普通的定时器

在软件方面，RemoDAQ-8188END 映射的 512K SRAM 存放在 512K Memory 低位空间；512K Flash 存放在 512K Memory 高位空间。软件通过 MiniBIOS，ROM-DOS 和 SOCKETS 存放，ROM-DOS 和 SOCKETS 都是很稳定的产品，Datalight 公司提供良好的客户支持服务。

2.1 内存映射

<table>
<tr><th>内存地址</th><th>内存状态</th><th>H/W 成分</th></tr>
<tr><td>0x00000~0x7FFFF</td><td>SRAM (512K)</td><td>SRAM 空</td></tr>
<tr><td>0x80000~0xEFFFF</td><td>Flash Memory (448K)
ROM DISK 始地址：0x80000</td><td rowspan="4">Flash 空</td></tr>
<tr><td>0xF0000~0xFBFFF</td><td>ROM DOS (48K)</td></tr>
<tr><td>0xFC000~0xFDFFF</td><td>保留（8K）</td></tr>
<tr><td>0xFE000~0xFFFFF</td><td>MiniBIOS（8K）</td></tr>
</table>

2.2 IO 映射

IO 地址	IO 状态	H/W 成分
0x000~0x0FF	Ad-on 卡	16C550s
0x100~0x2FF	保留	N/A
0x300~0x3FF	Ethernet 控制器	RTL8019
0x400~0x6FF	保留	N/A

2.3 MiniBIOS

MiniBIOS，像它的名字一样，是微型的 BIOS，在 ROM-DOS 中，它不是替代 BIOS，而是在不请求 BIOS 的情况下为内部环境服务。

MiniBIOS 作为 BIOS 的一个子集，在 ROM-DOS 中的操作和 BIOS 一样，包括 BIOS 支持的远程控制台、定时器、BIOS 扩展名、硬件设备等。MiniBIOS 不支持软驱、硬盘、打印机、键盘和鼠标。

名称	中断号	功能
Coprocessor ESC instruction	07H	保留
Timer 0 tick	08H	保留（硬件）
Keyboard Input	09H	高级电源管理
Serial receive char	0BH	保留（硬件）
Video TTY output	10H	0EH
Get equipment list	11H	
Get memory size in K	12H	
Disk I/O	13H	保留
Extended Memory Support	15H	保留
Keyboard Support	16H	00H，01H，02H
Keyboard,push Scan code into buffer	16H	05H
Boot failure message	18H	
System DOS-boot	19H	
Time of day	1AH	00H，01H
Timer Tick	1CH	

2.4 ROMDOS

RemoDAQ-8188END 模块中配有 ROM-DOS，ROM-DOS 是一个系统操作盘，能加载和完全运行 ROM，也能对硬盘或软盘进行操作。例如，在系统桌面上，ROM-DOS 和其他的 DOS 一样，在标准的 DOS 下运行程序，运行的程序可以存放在磁盘上或是在 ROM 中。

2.5 SOCKETS

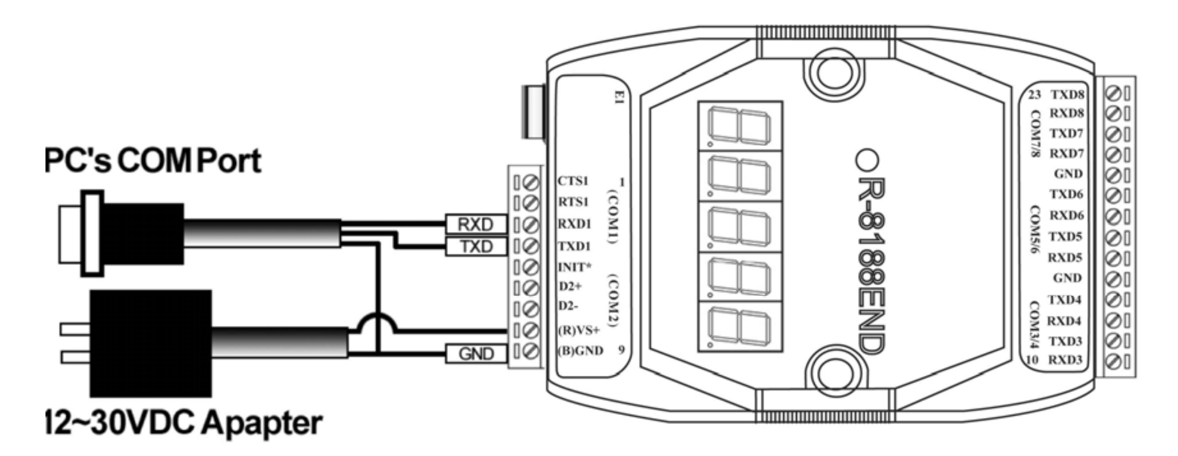
在嵌入式系统下运行 DOS，Datalight Sockets 是一个命令处理程序，即网络协议软件包。Datalight Sockets 还可在嵌入式系统下提供一个强大的数据通讯功能，用户可以和其他的计算机或打印机之间传输数据。

Datalight Sockets 是一个网络通讯应用软件，假若客户和服务器同时在网络上，你可以

- 从远程的嵌入式系统上通过主系统的 TCP/IP 运行应用软件
- 嵌入式系统和主系统的 TCP/IP 之间传输数据
- 在嵌入式系统上运行网络方面的软件
- 从主机的 TCP/IP 上打印嵌入式系统文件，反之亦然

3 快速入门

1. 连线说明：



2. RemoDAQ-8188END 的 COM1 与 PC 的 COM1(2)连接
3. 在 PC 的超级终端下（COM 端口设置：57600，8，N，1，无数据流程控制）
4. RemoDAQ-8188END 上电 (在 TCP 模式下)
   如果需要存放 PC 端口向 RemoDAQ-8188END 发送的命令，连接 INIT* pin 和 (B)GND 后并释放，RemoDAQ-8188END 将会在 ROMDOS 模式下。
5. PC 的超级终端显示 RemoDAQ-8188END 信息和 DOS 盘符 (A:\>)
6. 执行 DOS 命令，如 A:\ DIR
7. 传送或下载文件
8. 更新 ROMDISK 目录
9. 详见 S/W 用户手册中的函数说明，连接 ROM DISK，VDISK

<u>RemoDAQ-8188END配置</u>

用 INFOSET.EXE 命令读、写和设置模块
************RemoDAQ-8188END 有效信息设置************
M：修改模块　　NO：RemoDAQ-8188E1/E2/E3/E4/E5/E8
A：修改 MAC ADDR　0:10:f1:b6:7e:aa（只读）
I：修改 IP 地址　211.22.83.45
N：修改子网掩码　255.255.255.0
G：修改网关　192.9.40.120
B：修改波特率　1:57600 2:9600 3:9600 4:9600 5:9600 6:9600 7:9600 8:9600
D：修改数据位　1:8 2:8 3:8 4:8 5:8 6:8 7:8 8:8
P：修改奇偶校验　1:0 2:0 3:0 4:0 5:0 6:0 7:0 8:0
S：修改停止位　1:1 2:1 3:1 4:1 5:1 6:1 7:1 8:1
F：还原为默认值
Q：退出

4 工具

在 RemoDAQ-8188END 服务器上运行几个工具，包括以下部分
在超级终端环境下显示 A:\>
第一步：“开始 → 程序 → 附件 → 通讯 → 超级终端”
第二步：选择正确的 COM 端口，波特率（57600），数据位（8），奇偶（无），停止位（1）
第三步：RemoDAQ-8188END 上电，运行 XCOM.EXE 后，连接 INIT*和 GND 端，终端窗口显示 ROMDOS 盘符 A:\>

4.1 XM.EXE

在 PC 上显示程序，需要 XM.EXE 传输程序到 RemoDAQ-8188END 中来运行/测试。
下面是从 PC 向 RemoDAQ-8188END 上传文件的步骤：

1. 在 RemoDAQ-8188END 的 DOS 盘符下，运行“xm.exe /r <filename>”
2. 在超级终端窗口下选择：
   传送 →
   输出文件 →
   使用文件传输协议（Xmodem/1k-Xmodem） →
   文件名 →
   OK（60 秒内）
3. 文件上传完毕

或者从 RemoDAQ-8188END 向 PC 下载一个文件，步骤如下：

a．在 RemoDAQ-8188END 的 DOS 盘符下，运行“xm.exe /s <filename>”
b．在超级终端下选择：
   传送 →
   接收文件 →
   使用文件传输协议（1K-Xmodem） →
   放置文件(下载) →
   OK
c．文件下载完毕

4.2 ROMDISK.EXE

ROMDISK.EXE 是一个 ROM-DOS 工具，在 ROM-DOS 中 ROMDISK.EXE 是有效的，它可以建立一个 ROM-DOS 的图案，步骤如下：

1．在 ROM 中建立一个子目录
2．拷贝 ROMDISK 需要的所有文件
3．运行“ROMDISK.EXE”
4．通过“TOROM.EXE”下载文件到 RemoDAQ-8188END 模块中
5．重新启动模块，你会发现 RemoDAQ-8188END 模块的 A 盘中包含 ROM 所有文件目录

4.3 TOROM.EXE

TOROM.EXE 是用来传送 ROMDISK 的图像档案到 RemoDAQ-8188END 模块的 FLASH ROM 中。步骤如下：

1．连接 RemoDAQ-8188END 模块和 PC，使用超级终端作为终端
2．运行 TOROM.EXE
3．在超级终端下选择：传送 → 输出文件 →文件传输协议（Xmodem）→ 文件名 → OK
4．传输完毕后，重新启动 RemoDAQ-8188END 模块
5．你就可以看到更新后的 ROM DISK

注意：在 RemoDAQ-8188END 模块中执行 TOROM.EXE，第 3 步必须在 60 秒内完成，否则会发生时间溢出，在溢出的情况下，INIT*端子将下载程序到 FLASH ROM 中，如下：

RemoDAQ-8188END 模块的 INIT* pin 与 GND 连接、COM1 与 PC 连接完毕后系统上电，在超级终端下点击“传送”、“接收文件”，PC 上磁盘的图标将下载到 RemoDAQ-8188END 模块中，选择文件传输协议，键盘停留在文件名上，如果不成功，请重复 1~5 步。注意：在更新磁盘图标完成之前 INIT*和 GND 必须立即断开，否则，系统将报警。

4.4 VDISK

VDISK 是一个设备驱动程序，就像从存储系统中创建一个虚拟磁盘或 RAM DISK。断电后虚拟磁盘的目录将丢失。Vdisk.sys 是一个可以在 config.sys 里配置的系统文件，格式如下：
Device=vdisk.sys[size[secs[dirs]]]/E
Size 是 RAM DISK 空间的大小，默认为 64KB；Secs 是扇区的大小，默认是 512 字节，有效大小是 128，256，512；Dirs 是 RAM DISK 根目录下的目录和文件的数量，默认为 64。
例如：在 config.sys：
device=vidsk.sys
device=vdisk.sys 32 128 16
设置 64KB RAM DISK B:；512 字节扇区（当 RAM DISK A：存在）。
设置 32KB RAN DISK C:；128 字节扇区，16 个文件在根目录。

4.5 INFOSET.EXE

RemoDAQ-8188END 一些信息存放在 EEPROM 中，包括模块名称、MAC 地址、IP 地址、网关地址、COM 端口的数据格式，这些信息通过 INFOSET.EXE 很容易得到修改。

4.6 N.COM/SOCKET.EXE

N.COM 是 RemoDAQ-8188END 模块标准的信息包驱动，它为 SOCKETP.EXE 和 Ethernet 控制器提供界面，SOCKETP.EXE 是由 Datalight 为 SOCKETS/DOS TCP/IP 提供的内核。有两种工具必须执行才能允许 RemoDAQ-8188END 网络功能，他们是 NET.BAT，这就意味着你必须执行 RUN.BAT 才能允许 RemoDAQ-8188END 网络功能。

4.7 XPING.EXE

XPING.EXE 就像 PING.EXE 在 UNIX/WINDOWS 系统下一样，是最基本的查看网络是否正常工作的测试方式。它是由 DATALINGHT 提供的。

4.8 VSP/VIRCOM

过去通信应用是指一个物理通讯端口或系列相关连续的设备，现在变成了一个带有 VSP/VIRCOM 的 TCP/IP 网络。通过在 PC 上创建一个有效的端口，VSP/VIRCOM 改变有效的 COM 端口及数目和 IP 地址在系列服务器上的通讯即系列设备与网络的连接。并可在同一时间内为众多的设备服务。

使用有效的 COM 应用软件，1 台 PC 可以控制 255 个 COM 端口，VSP/VIRCOM 软件可以把 RemoDAQ-8188END 转换为 1 个 RS232Ethernet 转换器，执行通讯程序(接收和发送数据)就在 TCP/IP 端口上完成。用户很容易就能增强设置系统与 Ethernet/Internet 之间的连接，因为旧的 COM 上建立的程序不需要更改。

VSP/VIRCOM 首先需要安装驱动程序，然后，VSP/VIRCOM 工具能作用于 RemoDAQ-8188END 的任何一个端口并在 PC 上显现，PC 就可以在没有网络协议的情况下直接控制它们。VSP/VIRCOM 驱动程序将操作所有的 Ethernet/Internet 之间的连接。因此，用户不用担心网络连接。

4.9 XCOM.EXE

RemoDAQ-8188END 中 XCOM.EXE 是运行服务器程序，它接收由 PC 通过网络发出的命令并把他们传送到 RemoDAQ-8188END 中。RemoDAQ-8188END 成功启动后，XCOM.EXE 可设成 Autoexec.bat。从 NETTEST 中发送命令“00quit”就可以中断 XCOM.EXE 程序。

XCOM.EXE 运行时，5 位数字显示器顺序显示信息详见表 4.1

表 4.1

序号	格式	描述
1	11111	标签 1
2	1XAAA	X：空；AAA：1 个 IP
3	2XAAA	X：空；AAA：2 个 IP
4	3XAAA	X：空；AAA：3 个 IP
5	4XAAA	X：空；AAA：4 个 IP
6	22222	标签 2
7	1AAAA	AAAA：COM1 波特率
8	2AAAA	AAAA：COM2 波特率
9	3AAAA	AAAA：COM3 波特率
10	4AAAA	AAAA：COM4 波特率
11	5AAAA	AAAA：COM5 波特率

12	6AAAA	AAAA：COM6 波特率
13	7AAAA	AAAA：COM7 波特率
14	8AAAA	AAAA：COM8 波特率
15	33333	标签 3
16	1XABC	X：空；A：数据位；B：奇偶；C：COM1 停止位
17	2XABC	X：空；A：数据位；B：奇偶；C：COM2 停止位
18	3XABC	X：空；A：数据位；B：奇偶；C：COM3 停止位
19	4XABC	X：空；A：数据位；B：奇偶；C：COM4 停止位
20	5XABC	X：空；A：数据位；B：奇偶；C：COM5 停止位
21	6XABC	X：空；A：数据位；B：奇偶；C：COM6 停止位
22	7XABC	X：空；A：数据位；B：奇偶；C：COM7 停止位
23	8XABC	X：空；A：数据位；B：奇偶；C：COM8 停止位
24	XXXXX	XXXXX：空

XCOM.EXE 支持表 4.2 中的命令，你可以在 PC 上通过 NETTEST 测试它们，NETTEST 包括以下部分：

命令	说明	例子	
		输入	输出
00[arg]	停止服务器 XCOM.EXE Arg：退出 Return：无	00quit	无
01	读信息版本 Return：Vm.n.rr [mm/dd/yy] m：主版本 n：副版本 rr：返回数字	01	V1.0.00 [01/06/2004]
02[arg]	设置 COM 端口的波特率 Arg：NBBBB N：COM 端口（no1~8) BBBB: 设置的波特率 (1200,2400,4800,9600,19200,38400,57600,115200) Return: ERROR：Fail Ok：Success	02257600	OK
03[arg]	设置 COM 端口的数据格式 Arg：NDPS N：COM 端口（no1~8) D：数据位 （7,8==>COM1~COM2） (5,6,7,8==>COM3~COM8) P：奇偶校验 （0==>无校验） （1==>偶校验）	032801	OK

	（2==>奇校验） S：停止位 （1==>COM1~COM2） (1，2==>COM3~COM8) Return: ERROR：Fail Ok：Success		
06[arg]	设置 COM 端口的波特率，并在 SEPROM 中写出 Arg：NBBBB N：COM 端口 (no1~8) BBBB: 设置的波特率(1200,2400,4800,9600,19200,38400,57600,115200) Return: ERROR：Fail Ok：Success	0629600	OK
07[arg]	设置 COM 端口的数据格式，并在 EEPROM 中写出 Arg：NDPS N：COM 端口 (no1~8) D：数据位 （7,8==>COM1~COM2） (5,6,7,8==>COM3~COM8) P：奇偶校验 （0==>无校验） （1==>偶校验） （2==>奇校验） S：停止位 （1==>COM1~COM2） (1，2==>COM3~COM8) Return: ERROR：Fail Ok：Success	072801	OK
08[arg]	设置 IP 地址 Arg：iiipppIIIPPP iiipppIIIPPP：3 位数字（0~255） Return: ERROR：Fail Ok：Success	08192009040100	OK
10	读模块名称 Return：模块名称	10	RemoDAQ-8188END
11[arg]	测试服务器 Arg：SSS 测试字符串 Return: ERROR：Fail Ok：Success	11 Hello！	Hello！
12[arg]	设置网关地址 Arg：iiipppIIIPPP iiipppIIIPPP：3 位数字（0~255） Return: ERROR：Fail Ok：Success	12192009040254	OK

13	读取网关地址 Return： iii.ppp.III.PPP	13	192.9.40.254
14	设置掩码地址 Arg：iiipppIIIPPP iiipppIIIPPP：3 位数字（0~255） Return: ERROR：Fail Ok：Success	14255255255000	OK
15	读取掩码地址 Return： iii.ppp.III.PPP	15	255.255.255.0
16[arg]	读取 COM 端口设置 Arg：N N：COM 端口 (no1~8) Return：波特率，数据，奇偶校验，停止位	162	19200，8，0，1
20[arg]	禁止/允许 LED 显示信息 Arg：E E： 0=禁止 1=允许 Return: ERROR：Fail Ok：Success	201	OK
21	读 MAC 地址 Return: Xx：xx：xx：xx：xx：xx	21	00:00:F1:86:97:A6
28[arg]	重新设置 COM 端口 Arg：N N：COM 端口 (no1~8)	281	OK
99	Reboot R-8188END	99	

4.10 NETTEST

这是一个通过 TCP/IP 接收或发送数据到 RemoDAQ-8188END 中的工具。用户能列入 RemoDAQ-8188END 的 IP 地址，点击“CONNECT”连接端口，此时程序就可以向 RemoDAQ-8188END 发送命令、接收数据。“NETTEST”主要是用来判断 RemoDAQ-8188ENDEthernet 的连接。

在 NETTEST 目录中双击“setup.exe”，即可安装，图 4.1 是显示 NETTEST 的 GUI，下面是操作中的一些技巧：

- 连接端口，在“Server setting”输入 RemoDAQ-8188END 的 IP
- 点击“Connect”，如果连接成功，“Connect”将会变成“Disconect”，服务器的信息显示在“Server Information”中
- 点击“Commands”，在命令表中选择要发送的命令，直接回车或点击“Send” RemoDAQ-8188END 的信息将会出现在“R-8188E Response Window”中
- 点击“Send”，向 COM 端口发送字符串
- 回答将出现在“Receive”中

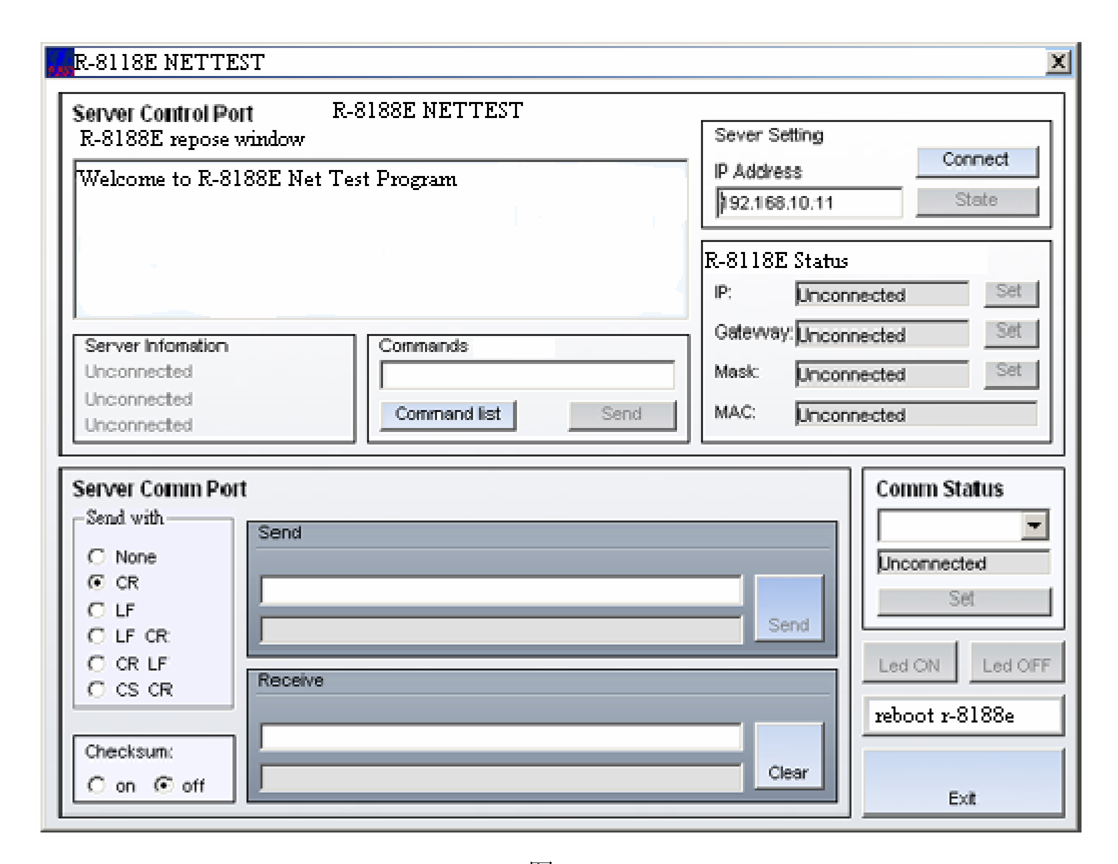


图 4.1

用 TCP/IP 连接，TCP10000 端口只是用来交换命令，10000N 端口是属于 COMN 的其他数据之间的传输。举例如下：

选择与 RS485 设备或者是 Remo8188END 中 RemoDAQ8000 系列连接的 COM 端口，发送命令：

- 命令： $01M　　接收： !8041D
- 命令： $012　　接收： !01400600
- 命令： $01F　　接收： !VER 1.A

5 RemoDAQ-8188END 函数库

5.1 节~5.6 节的函数库与 RemoDAQ-8188END 具有一定的兼容性，详情请查阅 RemoDAQ-8188END 的库函数用户指南，5.7 节 Socket API 更多的详情请查阅 Datalight Sockets2.0 开发指南，这里只是介绍一些函数原型，仅供参考。

5.1 FLASH

```
int flashRead (void);
int FlashWrite (unsigned int seg,unsigned int offset,char data);
int FlashErase (unsigned int seg);
int FlashRead (unsigned int seg,unsigned int offset);
```

5.2 NVRAM/SERIAL EPROM

```
int ReadNVRAM (int addr);
int WriteNVRAM (int addr,int data);

int WriteEEP (int block,int addr,int data);
int ReadEEP (int block,int addr);
void EnableEEP (void);
void ProtectEEP (void);
```

5.3 LED/5 位数字显示器

```
void LedOff (void);
void LedOn (void);

void init5DigitLed (void);
int Show5DigitLed (int position,int value);
int Show5DigitLedSeg (int pos,int value);
int Show5DigitLedWithDot (int pos,int data);
void Set5DigitTestMode (int mode);
void Set5DigitLedintensity (input mode);
void Disable5DigitLed (void);
void Enable5DigitLed (void);
```

5.4 COM 端口

int InstallCom (port,unsigned long baud,int data,int parity,int stop);
int RestoreCom (int port);
int IsCom (int port);
int ReadCom (int port);
int ToCom (int port);
int ClearCom (int port);
int WaitTransmitOver (int port);
int DataSizeInCom (int port);
void SetFlowControActive (void);
void SetFlowControInactive (void);
int SendCmdTo7000 (int iPort,unsigned char *cCmd,int iChksum);
int RecevieResponseFrom8000 (int iPort,unsigned char *cCmd,long 1timeout,int iChksum);

5.5 定时器

int TimerOpen (void);
int TimerClose (void);
void TimerResetValue (void);
unsigned long TimerReadValue (void);
void DelayMs (unsigned t);
int StopWatchReset (int channel);
int StopWatchStart (int channel);
int StopWatchStop (int channel);
int StopWatchPause (int channel);
int StopWatch Continue (int channel);
int StopWatchReadValue (int channel,unsigned long *value);
int CountDownTimerStart (int channel,unsigned long count);
int CountDownTimerReadValue(int channel,unsigned long *value);
void InstallUserTimer (void (*fun)(void));
void InstallUserTimer1C (void (*fun)(void));
void Delay (unsigned ms);
void Delay_1 (unsigned ms);

5.6 ETC

int getch4 (void);
int kbhit4 (void);

```
int ungetch4 (int key);
void putch4 (int data);
int ReadInitPin (void);
```

5.7 SOCKET API

```
int DisableAsynvNotification (void);
int EnableAsyncNotification (void);
DWORD GetAddress (int isocket);
Int GetPeerAddress (int iSocket,NET_ADDR *psAddr);
Int GetKeme1Information (int iReserved,BYTE bCode,BYTE bDevID,void
                        *pData,Word *pwSize);
int GetVersion (void);
int ICMPPing (DWORD dwHost,int iLength);
int IsSocket (int iSocket);
int GetDCSocke (void);
int GetSocket (void);
int GetKernel1Config (KERNEL_CONFIG *psKC);
int ConvertDCSocket (int iSocket);
int GETNetInfo (int iSocket,NET_INFO *psNI);
int ConnectSocket (int iSocket,int iType,NET_ADDR *psAddr);
int ListenSocket (int iSocket,int iType, NET_ADDR *psAddr);
int SelectSocket (int iMaxID,long *p1Iflags,long *p1Oflags);
int ReadSocket (int iSocket,char *pcBuf,WORD wLen,
                        NET_ADDR *psBuf,WORD wFlags);
int ReadFromSocket (int iSocket,char *pcBuf,WORD wLen,
                        NET_ADDR *psBuf,WORD wFlags);
int WriteSocket (int iSocket,char *pcBuf,WORD wLen,WORD wFlags);
int WriteToSocket (int iSocket,char *pcBuf,WORD wLen,
                        NET_ADDR *psBuf,WORD wFlags);
int Eofsocket (int iSocket);
int FlushSocket (int iSocket);
int ReleaseSocket (int iSocket);
int ReleaseDCSockets (void);
int AbortSocket (int iSocket);
int AbortDCSockets (voidt);
int ShutDownNet (void);
int SetAlarm (int iSocket,DWORD dwTime,int (D_FAR *1pHandler()0,DWORD dwHint);
int D_FAR *SetAsyncNotification (int iSocket,int iEvent,
                        int (D_FAR *1pHandler)0,DWORD dwhint);
```

DWORD ResolveName (char *pszName,char *pcCName,int iCnameLen);
DWORD ParseAddress (char *pszName);
int SetSocketOption (int iSocket,int iLevel,int iOption,DWORD dwOption Value,int iLen);
int JoinGroup (DWORD dwGroupAddress,DWORD dwInterfaceAddress);
int LeaveGroup (DWORD dwGroupAddress,DWORD dwInterfaceAddress);
int IfaceIOCTL (char *pszName,WORD wFunction);
int GetSocketsVersion (void);